UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/007,582	12/05/2001	Roy F. Brabson	RSW920010222US1	3561

46589 7590 10/10/2007
MYERS BIGEL SIBLEY SAJOVEC P.A.
PO BOX 37428
RALEIGH, NC 27627

EXAMINER	
PAN, JOSEPH T	

ART UNIT	PAPER NUMBER
2135	

MAIL DATE	DELIVERY MODE
10/10/2007	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

UNITED STATES PATENT AND TRADEMARK OFFICE

Commissioner for Patents
United States Patent and Trademark Office
P.O. Box 1450
Alexandria, VA 22313-1450
www.uspto.gov

**BEFORE THE BOARD OF PATENT APPEALS
AND INTERFERENCES**

Application Number:	10/007,582
Filing Date:	Dec. 5, 2001
Appellant(s):	Brabson et al.

D. Scott Moore
For Appellant (Reg. No: 42,011)

EXAMINER'S ANSWER

This is in response to the Appeal brief filed on June 4, 2007 appealing from the Office action mailed on January 4, 2007.

(1) Real Party in Interest

A statement identifying by name the real party in interest is contained in the brief.

(2) Related Appeals and Interferences

The examiner is not aware of any related appeals, interferences, or judicial proceedings which will directly affect or be directly affected by or have a bearing on the Board's decision in the pending appeal.

(3) Status of Claims

The statement of the status of claims contained in the brief is correct.

(4) Status of Amendments After Final

The appellant's statement of the status of amendments after final rejection contained in the brief is correct.

(5) Summary of Claimed Subject Matter

The summary of claimed subject matter contained in the brief is correct.

(6) Grounds of Rejection to be Reviewed on Appeal

The appellant's statement of the grounds of rejection to be reviewed on appeal is correct.

(7) Claims Appendix

The copy of the appealed claims contained in the Appendix to the brief is correct.

(8) Evidence Relied Upon

US 6,370,599 B1 Anand et al. Jun. 12, 1998

(9) Grounds of Rejection

The following ground(s) of rejection are applicable to the appealed claims:

1. Claims 1-20 are rejected under 35 U.S.C. 102(e) as being anticipated by Anand et al. (U.S. Patent No. 6,370,599 B1), hereinafter referred to as Anand.

Referring to claim 1:

Anand teaches:

A method of improving security processing in a computing network, comprising:

Providing a security offload component in an operating system kernel which performs security processing (see figure 2; and column 3, lines 13-60 of Anand);

Providing a control function in an operating system kernel for directing operation of the security offload component (this security offload component is interpreted "Figures 9 through 14 provide flowcharts which illustrate logic that may be used to implement an embodiment of the present invention which performs secure data transfer offload; and Figures 15 through 17 provide message flow diagrams showing message exchanges that may be used to implement secure handshake offload, according to another embodiment" which is in according to applicant's description on page 11, lines 13-19 of the Specification, see column 3, lines 32-36; and column 15, lines 40-56 of Anand);

Providing an application program (see abstract, lines 20-28 of Anand);

Executing the application program (see abstract, lines 20-28 of Anand); and

Executing the provided control functions during execution of the application program, thereby directing the security offload component to secure at least one communication of the executing application program (see abstract, lines 20-28 of Anand).

Referring to claim 2:

Anand teaches the claimed subject matter: a method of improving security processing in a computing network (see claim 1 above). Anand further discloses directing the security offload component to begin securing the communications (see column 3, lines 32-36 of Anand).

Referring to claim 3:

Anand teaches the claimed subject matter: a method of improving security processing in a computing network (see claim 1 above). Anand further discloses directing the security offload component to stop securing the communications (see abstract, lines 26-28 of Anand).

Referring to claim 4:

Anand teaches the claimed subject matter: a method of improving security processing in a computing network (see claim 1 above). Anand further discloses specifying information to be used by the security offload component (see column 10, lines 43-63 of Anand).

Referring to claim 5:

Anand teaches the claimed subject matter: a method of improving security processing in a computing network (see claim 1 above). Anand further discloses the specified information including the specified encryption key, and other predefined data (see column 10, line 64 to column 11, line 12 of Anand).

Referring to claims 6-7, 16, 20:

Anand teaches the claimed subject matter: a method of improving security processing in a computing network (see claim 1 above). Anand further discloses modifying outbound data in preparation for use by the security offload component (see column 10, lines 43-63 of Anand).

Referring to claim 8:

Anand teaches the claimed subject matter: a method of improving security processing in a computing network (see claim 1 above). Anand further discloses the certificates (see column 2, lines 55-60; and column 10, line 64, to column 11, line 12 of Anand).

Referring to claim 9:

Anand teaches the claimed subject matter: a method of improving security processing in a computing network (see claim 1 above). Anand further discloses the encryption key (see column 10, line 64 to column 11, line 12 of Anand).

Referring to claim 10:

Anand teaches the claimed subject matter: a method of improving security processing in a computing network (see claim 1 above). Anand further discloses the encryption algorithm (see column 10, lines 2-4 of Anand).

Referring to claim 11:

Anand teach the claimed subject matter: a method of improving security processing in a computing network (see claim 1 above). Anand further disclose that the secured outbound data of the executing application is thereby sent to its destination directly from the security offload component, after a single path over a data bus from a protocol stack of the operating system (see figure 3; and abstract, lines 20-28 of Anand).

Referring to claim 12:

Anand teaches:

A system for improving security processing in a computing network, comprising:

A security offload component in an operating system kernel which performs security processing (see figure 2; and column 3, lines 13-60 of Anand);

At least one control function in the operating system kernel for directing operation of the security offload component (see column 3, lines 32-36 of Anand);

Means for executing the at least one provided control function (see abstract, lines 20-28 of Anand); and

Means, responsive to operation of the means for executing, for directing the security offload component to secure at least one communication of an application program (see abstract, lines 20-28 of Anand).

Referring to claim 13:

Anand teaches:

A computer program product for improving security processing in a computing network, the computer program product embodies on at least one computer-readable media and comprising:

A security offload component in an operating system kernel which performs security processing (see figure 2; and column 3, lines 13-60 of Anand);

At least one control function in the operating system kernel for directing operation of the security offload component (see column 3, lines 32-36 of Anand);

Computer-readable program code for executing the at least one provided control function (see column 3, lines 32-36 of Anand); and

Computer-readable program code, responsive to operation of the computer-readable program code for executing, for directing the security offload component to secure at least one communication of an application program (see abstract, lines 20-28 of Anand).

Referring to claims 14, 18:

Anand teaches the claimed subject matter: a system of improving security processing in a computing network (see claim 12 above). Anand further discloses directing the security offload component to begin securing the communications (see column 3, lines 32-36 of Anand).

Referring to claims 15, 19:

Anand teaches the claimed subject matter: a system of improving security processing in a computing network (see claim 12 above). Anand further discloses directing the security offload component to stop securing the communications (see abstract, lines 26-28 of Anand).

Referring to claim 17:

Anand teaches the claimed subject matter: a system of improving security processing in a computing network (see claim 12 above). Anand further discloses that the secured outbound data of the executing application is thereby sent to its destination directly from the security offload component, after a single path over a data bus from a protocol stack of the operating system (see figure 3; and abstract, lines 20-28 of Anand).

(10) Response to Argument

A. Independent Claims 1, 12 and 13 are patentably over Anand.

i. "As highlighted above, independent Claims 1, 12, and 13 include recitations directed to providing a security offload component in an operating system kernel. That is, the security offload component is provided as part of the Operating system kernel software. In sharp contrast, Anand is directed to moving tasks that are typically performed in software to a hardware component." (see page 5, last paragraph)

Anand first discloses a functional block diagram illustrating the flow of a data packet through program components (see figure 3, element 128 'transport protocol driver, e.g. TCP/IP', 144 'data packet functions, e.g., IP security driver', 116 'network driver', 100 'NIC hardware e.g., ethernet', of Anand), wherein "in the Windows NT layered networking architecture, a transport protocol driver, or transport, is implemented with an appropriate program method so as to be capable of querying each of the device driver(s) associated with the corresponding NIC(s) connected to the computer." (see column 3, lines 61-66 of Anand). Thus, Anand discloses that the transport protocol driver[figure 3, element 128 of Anand], or transport, is implemented in the Windows NT layered networking architecture [i.e., Windows NT Operating System Kernel].

Anand then discloses "in FIG. 3, the data packet 142 is passed to a **software component 144**, which could be implemented separately or implemented as a part of the transport protocol driver itself, that **appends a packet extension** to the packet 142. Data will be included within in packet extension **depending on the particular task that is to be offloaded**." (see column 10, lines 64-column 11, line 2 of Anand, emphasis added).

Anand further discloses "Each NIC is logically interconnected with the Windows NT networking model, as is schematically represented by bidirectional lines 108-112, via a corresponding network driver 116-120. Network drivers reside in the MAC sublayer of the network model, and link Windows NT to the physical network channel via the corresponding NICs. Each driver, typically implemented as a software component provided by the vendor of the corresponding NIC, is responsible for sending and receiving packets over its corresponding network connection and for managing the NIC on behalf of the operating system. Each driver also starts I/O on the corresponding NIC and receives interrupts from them, and calls upward to protocol drivers to notify them of its completion of an outbound data transfer. Also, the device driver will be responsible for invoking, controlling and/or monitoring any of the additional processing capabilities of the corresponding NIC." (see column 8, lines 36-52 of Anand, emphasis added).

Thus, the IP Security Driver 144 in Anand is operating in the Windows NT having data structure field appended to the data packet for identifying particular tasks, i.e., encryption(security processing), checksum, etc.,being offloaded.

Independent claims 12 and 13 cited similar limitation to that of claim 1. Therefore, the same response to argument above is applied to claims 12 and 13.

ii. "Appellants agree that the Specification describes several embodiments in which a security offload component is implemented as a hardware device. But as highlighted above, the Specification also describes numerous embodiments in which security processing is offloaded from the application into the operating system kernel, such as the TCP layer, as discussed with reference to FIGS. 2A through 2E." (see page 6, last paragraph)

See the same response to argument (i) above, particularly note figure 3, element 128 'transport protocol driver, e.g., TCP/IP', element 144 'data packet functions, e.g., IP security driver' of Anand.

Therefore, Anand discloses that security processing is offloaded from the application into the operating system kernel, such as the TCP layer.

iii. "Appellants further submit that Anand inherently does not disclose or suggest the recitation "providing control functions in the operating system kernel for directing operation of the security offload component" of Claim 1 and analogous recitations of Claims 12 and I3 as Anand describes moving the security functionality into a hardware component, such as a NIC, to relieve the CPU, which executes the operating system software, of that task." (see page 7, 2nd paragraph)

Anand discloses "in the Windows NT layered networking architecture, **a transport protocol driver, or transport**, is implemented with an appropriate program method so as to be capable of querying each of the device driver(s) associated with the corresponding NIC(s) connected to the computer. Each queried device driver is similarly implemented so as to be capable of responding by identifying its specific processing, or "task offload" capabilities. In a preferred embodiment, once the task offload capabilities of each individual peripheral device have been identified, the **transport** sets which of those specific capabilities are to be enabled. This essentially informs the peripheral device what type of tasks it should expect to perform during subsequent transmissions and/or receptions of data packets. Thereafter, the **transport** is able to take advantage of the enabled capabilities of a peripheral device on an as-needed basis. Preferably, the enabled functions are invoked via appropriate data that is appended to the actual data packet destined for the network channel. In this way, tasks can be offloaded dynamically, and more than one task can be offloaded at a time." (see column 3, lines 61-column 4, lines 14 of Anand).

Anand further discloses "a step for ascertaining, **by the operating system**, task offload capabilities of the peripheral hardware device; a step for enabling, **by the operating system**, selected task offload capabilities of the peripheral hardware device that are selected from among the ascertained task offload capabilities, said selected task offload capabilities being enabled to the extent such selected task offload capabilities are needed for one or more data packets;" (see column 15, lines 48-56 of Anand)

Therefore, Anand discloses "providing control functions in the operating system kernel for directing operation of the security offload component".

B. Dependent Claims 8-11 and 17 are patentably over Anand.

i. "Appellants submit that Anand does not appear to disclose any of the specific details of dependent Claims 8-10, and 17." (see page 7, 5th paragraph)

Anand discloses "in FIG. 3 the application data 140 is passed down from the upper layers of the network model to an appropriate transport protocol driver, such as TCP/IP 128. The driver repackages the data into an appropriate data packet 142. Then, depending on whatever additional functions are to be performed on this particular data packet 142 a functional component is included that appends a predefined data structure, referred to as the packet extension, to the data packet. As will be discussed in further detail below, the contents of the packet extension reaches the NIC 100. When the data packet 142 reaches the network driver 116, the contents of this packet extension are queried by the driver 116 so as to ascertain which task(s) is to be performed by the NIC 100. **The driver** 116 then **controls/manipulates** the **hardware** on the NIC so that it will perform whatever functional tasks have been requested via the **contents of the packet extension**." (see column 10, lines 46-63 of Anand, emphasis added). In figure 4, Anand further discloses that the packet extension 150 includes IPSec info (figure 4, element 154), and IPSec info contains various fields related to IPSec [e.g. IPSec certificate].

Anand further discloses "For example, in FIG. 3, the data packet 142 is passed to a software component 144, which could be implemented separately or implemented as a part of the transport protocol driver itself, that appends a packet extension to the packet 142. Data will be included within in packet extension depending on the particular task that is to be offloaded. For instance, if an IP security function is to be implemented, data that indicates that the NIC 100 should encrypt the data packet in accordance with a specified encryption key would be included." (see column 10, line 64-column 11, line 6 of Anand, emphasis added). Anand further discloses the cryptographic algorithm (see column 10, lines 2-3 of Anand).

Anand then discloses "Once an application has discerned the capabilities of a particular NIC, it will selectively utilize any of the enabled task offload capabilities of the NIC by appending packet extension data to the network data packet that is forwarded to the NIC. The device driver of the NIC will review the data contained in the packet extension, and then cause the NIC to perform the specified operating task(s).This offloading of computing tasks on a per-packet basis allows an application to selectively offload tasks on a dynamic, as-needed basis." (see abstract, lines 20-28 of Anand). Figure 3 illustrates the secured outbound data of the executing application is sent to its destination directly from the security offload component, after a single path over a data bus from a protocol stack of the operating system.

Therefore, Anand discloses the specific details of dependent Claims 8-10, and 17.

(11) Related Proceeding(s) Appendix

No decision rendered by a court or the Board is identified by the examiner in the Related Appeals and Interferences section of this examiner's answer.

For the above reasons, it is believed that the rejections should be sustained.

KIM VU
SUPERVISORY PATENT EXAMINER
TECHNOLOGY CENTER 2100

Respectfully submitted,

Joseph Pan JP

Conferees:

Ho Song HS.

Kim Vu KV

HOSUK SONG
PRIMARY EXAMINER